

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Амурский государственный университет»

Кафедра математического анализа и моделирования

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Практикум на ЭВМ

Основной образовательной программы по специальности 010501.65 – «Прикладная математика и информатика»

Благовещенск 2012 г.

УМКД разработан доцентом Труфановым Виктором Александровичем

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от « 11 » января 2012 г. № 5

Зав. кафедрой _____ / В.В. Сельвинский /

УТВЕРЖДЕН

Протокол заседания УМСС 010501.65 – Прикладная математика и информатика

« 11 » января 2012 г. № 5

Председатель УМСС _____ / В.В. Сельвинский /

СОДЕРЖАНИЕ

I	Рабочая программа учебной дисциплины	
1	Цели и задачи изучения дисциплины	4
2	Место дисциплины в структуре ООП ВПО	4
3	Структура и содержание дисциплины «Практикум на ЭВМ»	4
4	Содержание разделов и тем дисциплины	5
5	Самостоятельная работа	7
6	Образовательные технологии	7
7	Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов	7
8	Учебно-методическое и информационное обеспечение дисциплины	8
9	Материально-техническое обеспечение дисциплины	8
II	Лабораторный практикум: методические указания по выполнению лабораторных работ или комплект заданий	
2.1	Программирование на языке высокого уровня C/C++.	9
2.2	Среда Delphi и Object Pascal.	11
2.3	Применение математических пакетов Mathcad и Matlab для решения задач линейной алгебры, математического анализа и численных методов.	13
2.4	Основы программирования трехмерной графики с использованием библиотеки OpenGL	14
2.5	Динамические типы данных и объектно-ориентированное программирование	16
III	Контроль знаний	
3.1	Фонд тестовых и контрольных заданий для оценки качества знаний по дисциплине	18
3.2	Вопросы к зачету.	21

I РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

1. ЦЕЛЬ И ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

1.1. Цель преподавания дисциплины

Данный курс составляет часть дисциплин: информатика, языки программирования, методы оптимизации, численные методы. Поэтому основной целью «Практики на ЭВМ» является приобретение практических навыков применения ЭВМ при решении задач различного типа через средства программируемых сред для разработки программ (программных комплексов или алгоритмов), используя численные методы.

1.2. Задачи изучения дисциплины:

- освоение среды программирования;
- освоение основных принципов разработки алгоритмов и программ;
- использование простейших элементов и объектов программных средств для разработки программных комплексов;
- изучение и использование различных методов программирования;
- создание и разработка алгоритмов решения задач;
- отладка программ по заранее готовым тестам;
- разработка рабочих программ, использующих классические численные методы, в том числе численные методы линейной алгебры, численные методы мат. анализа, методы оптимизации; методы решения дифференциальных. уравнений;
- модифицировать алгоритмы и методы решения в соответствии с условиями задач;
- получение навыков оформления программной документации.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

ОПД.Ф.7 по дисциплине «Практикум на ЭВМ»:

практикум по программированию; практикум по решению прикладных задач (практическое освоение работы на ЭВМ, умение применять стандартные математические методы и математическое обеспечение ЭВМ для решения различных задач).

Дисциплина является вспомогательной для изучения других дисциплин специальности, связанных с программированием.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 436 часа.

Дисциплина рассчитана на пять семестров. Во втором семестре раскрываются общие концепции языков высокого уровня. Подробно рассматривается язык высокого уровня C/C++. Уделяется особое внимание алгоритмизации и структурному подходу к написанию программ. Делается акцент на операции ввода-вывода.

В третьем семестре изучается среда программирования Delhi. Детально рассматриваются свойства объектов с точки зрения программирования на языке Object Pascal.

В четвертом семестре рассматриваются прикладные математические задачи, которые решаются с помощью пакетов Mathcad, Matlab.

В пятом семестре изучается язык маркировки гипертекстов HTML, разработанный для создания WEB-страниц с соблюдением принципа независимости от аппаратных платформ.

В шестом семестре рассматривается объектно-ориентированный подход к написанию программ, динамическое программирование.

В результате освоения курса студенты должны уметь самостоятельно определять способы и методы решения поставленной задачи, строить алгоритм решения и разрабатывать программу, реализующую построенный алгоритм, в соответствии с правилами и

принципами построения программы на языке (C/C++, Паскаль, Delphi). Студент также должен получить опыт работы на персональном компьютере (ПК).

5. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

Темы лабораторных занятий

II семестр. Программирование на языке высокого уровня C/C++.		
	Темы лабораторных работ	Кол-во часов
1.	Обучение работы в интегрированной среде Visual C++ 6.	2 ч.
2.	Линейные алгоритмы.	4 ч.
3.	Разветвляющиеся, циклические и итерационные алгоритмы.	14 ч.
4.	Одномерные массивы и указатели.	8 ч.
5.	Двумерные массивы.	8 ч.
6.	Строки и файлы.	8 ч.
7.	Структуры.	8 ч.
8.	Функции.	6 ч.
9.	Рекурсия.	6 ч.
10.	Элементарная машинная графика.	6 ч.
11.	Зачетное занятие.	2 ч.
	Итого	72 ч.

III семестр. Среда Delphi и Object Pascal.		
	Темы лабораторных работ	Кол-во часов
1.	Среда разработки Delphi.	2 ч.
2.	Использование визуальных компонентов:	
	+ компоненты для обработки текста;	4 ч.
	+ компоненты управляющих команд;	4 ч.
	+ графические компоненты;	2 ч.
	+ компоненты для доступа к файлам и каталогам.	4 ч.
3.	Создание приложений:	
	+ с использованием меню;	4 ч.
	+ с использованием диалоговых окон;	4 ч.
	+ с использованием графических объектов;	6 ч.
	+ с использованием главной и дочерней форм.	4 ч.
4.	Использование различных моделей программирования: инкапсуляция, полиморфизм, наследование объектов.	6 ч.
5.	Использование методов при программировании: виртуальные, динамические, абстрактные.	6 ч.
6.	Работа с графикой.	6 ч.
7.	Зачетное занятие.	2 ч.
	Итого	54 ч.

IV семестр. Применение математических пакетов MathCAD и MATLAB для решения задач линейной алгебры, математического анализа и численных методов.		
	Темы лабораторных работ	Кол-во часов
1.	Пакет Mathcad – основные возможности.	2 ч.
2.	Графические возможности. Создание различного рода графиков, их форматирование и использование.	8 ч.
3.	Символьные вычисления.	2 ч.
4.	Действия с векторами и матрицами.	8 ч.
5.	Решение алгебраических выражений и систем.	6 ч.
6.	Аппроксимация функций.	6 ч.
7.	Решение дифференциальных уравнений.	6 ч.
8.	Программирование.	6 ч.
9.	Контрольное занятие.	2 ч.
10.	Основы работы с пакетом MATLAB.	6 ч.
11.	Зачетное занятие.	2 ч.
	Итого	54 ч.

V семестр. Основы программирования на HTML, Java Script.		
	Темы лабораторных работ	Кол-во часов
1.	Основные сведения языка HTML	2
2.	Основные теги, предназначенные для работы с текстом.	6
3.	Графические файлы в HTML-документе.	6
4.	Фон веб-страниц.	4
5.	Линии, точки, Java-апплеты, сценарии и другие объекты.	8
6.	Бегущая строка на веб-странице.	4
7.	Виды списков. Таблицы.	6
8.	Формы. Ссылки. Слои.	6
9.	Каскадные листы стилей. Баннерная реклама.	6
10.	Подготовка изображений для веб-страниц	6
11.	Основы Java Script.	18
	Итого	72

VI семестр. Динамические типы данных и объектно-ориентированное программирование.		
	Темы лабораторных работ	Кол-во часов
1.	Работа с динамическими переменными.	2 ч.
2.	Односвязные списки (модификация мини ИС).	4 ч.
3.	Двусвязные списки (доработка мини ИС).	4 ч.
4.	Стеки, стековый калькулятор.	6 ч.
5.	Очереди, задачи моделирования очередей.	4 ч.
6.	Двусвязное кольцо.	4 ч.
7.	Деревья.	4 ч.
8.	Применение ООП.	4 ч.
9.	Использование инкапсуляции.	4 ч.
10.	Использование наследования.	4 ч.
11.	Использование полиморфизма.	4 ч.
12.	Создание динамических объектов.	6 ч.
13.	Зачетное занятие.	2 ч.
	Итого	54 ч.

6. САМОСТОЯТЕЛЬНАЯ РАБОТА

Поскольку курс не имеет лекционных часов и базируется на теорию других дисциплин (см. выше), то для самостоятельной работы предлагается:

- 1) разработка тестов для отладки программ;
- 2) изучения и соответствующих способов и методов решения задач, условий существования решения и анализ полученных результатов;
- 3) использовать рекомендованную литературу.

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Учебные занятия: лабораторные занятия, консультации, самостоятельная работа.

9. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

В течение семестра студенты разбирают и решают задачи, указанные преподавателем к каждой лабораторной работе. Осуществляется защита выполненного задания в виде вопросов. Например, таких:

1. Какие типы величин используются в языке C++?
2. Указать диапазон значений величин типов int и float.
3. Какие имена переменных допустимы в программе?
4. Можно ли в качестве операнда в арифметическом выражении использовать: а) имя стандартной функции; б) имя массива; в) имя символьной переменной или переменной логического типа?
5. Что напечатает следующий оператор cout << “*\n**\n***\n****\n*****\n”?
6. Перечислить действия, реализуемые при выполнении условного оператора.
7. Что такое вычислительный процесс разветвляющейся структуры?
8. Как организовать разветвление вычислений а) на две ветви; б) на три ветви?
9. Зачем необходимо при отладке программы тестировать все ветви алгоритма?
10. Указать последовательность действий, выполняемых при организации циклических участков программы с заданным числом повторений
11. Что такое итерационный циклический процесс? Его отличия от цикла с заданным числом повторений.
12. Какие условия сходимости метода итераций?
13. Каково условие выхода из цикла при вычислении значения суммы бесконечного ряда?
14. Зачем используются рекуррентные соотношения для вычисления значений ряда?
15. Какие операторы можно использовать для описания массивов?
16. Возможен ли выход из внутреннего цикла до его полного завершения, досрочное завершение работы программы?
17. В каких случаях можно не освобождать память, выделенную посредством new []?
18. Указать основные правила организации вложенных циклов?
19. Как добавить элемент в односвязный список?
20. Основные отличия динамического выделения памяти, от статического?
21. Особенности арифметических операций в MatLab.
22. Как осуществляется связь компонент формы, с программным кодом в Delphi?
23. Как осуществляется работа с графикой в Delphi?

10. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ «Практикум на ЭВМ»

а) основная литература:

1. Бобровский С. И. Delphi 7: учеб. курс - СПб. : Питер , 2004, 2006. - 736 с.
2. Зубов В.С. Object Pascal: практикум в среде DELPHI /. - М. : Изд-во Моск. энергет. ин-та, 2004. - 272 с.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2010.
4. Ракидин В.И. Руководство по методам вычислений и приложения MATHCAD: учеб. пособие: доп. УМО - М. : Физматлит, 2005. - 264 с.
5. Труфанов В.А., Рыженко А.В. С++. Начало программирования: лабораторный практикум. – Благовещенск.: Амурский гос. у-т, 2008. – 144 с.
6. Черняк А.А. Высшая математика на базе Mathcad. Общий курс Учеб. пособие / А. А. Черняк , Ж. А. Черняк, Ю. А. Доманова. - СПб. : БХВ-Петербург, 2004. - 593 с.

б) дополнительная литература:

1. Баженова И. Ю. Delphi 5. Самоучитель программиста: учеб.- справ. пособие / Баженова И.Ю. - М. : Кудиц-Образ, 2000. - 336с.
2. Галаган Т. А. Язык программирования С++ в примерах и задачах: учеб. - метод. пособие / Т. А. Галаган, Л. А. Соловцова; АмГУ, ФМиИ. - Благовещенск : Изд-во Амур. гос. ун-та, 2002. - 148 с..
3. Кэнту Марко. Delphi 7. - СПб. : Питер, 2005. - 1101 с.
4. Павловская Т.А. С/С++. Структурное программирование: практикум / Т.А. Павловская, Ю.А. Щупак. - СПб. : Питер , 2004. - 239 с.
5. Решетнева Т. Г. Алгоритмизация и программирование в среде Mathcad: практикум; - Благовещенск : Изд-во Амур. гос. ун-та, 2002. - 77 с.

в) программное обеспечение: интегрированные среды С++, Delphi, математические пакеты Mathcad и Matlab.

г) Интернет-ресурсы:

№	Наименование ресурса	Краткая характеристика
1	http://www.iqlib.ru	Интернет-библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знания

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лаборатории численных методов исследования динамических систем (327 ауд.) или компьютерного моделирования и эксперимента (329 ауд.), оснащенные восемью ПК Pentium 2,49 ГГц с ОС Microsoft Windows XP.

II. ЛАБОРАТОРНЫЙ ПРАКТИКУМ: МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ИЛИ КОМПЛЕКТ ЗАДАНИЙ

Задания для лабораторных, контрольных работ и домашних заданий берутся из книг, реквизиты которых приведены в рабочей программе.

2.1 Программирование на языке высокого уровня C/C++ 2 семестр

Лабораторная работа № 1. Программирование алгоритмов линейной структуры.

Цель работы. Овладение практическими навыками разработки и программирования вычислительного процесса линейной структуры и навыками по отладке и тестированию программ.

Задания для самостоятельной подготовки.

1. Изучить:

- запись констант, переменных, стандартных функций;
- правила записи арифметических выражений;
- организацию простейшего ввода-вывода данных.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи.

4. Подготовить тестовый вариант исходных данных и вычислить для них с помощью калькулятора значения вычисляемых в программе величин, затем следует сравнить их со значениями, вычисленными в компьютере (результат вычисления по первой формуле должен совпадать со второй, но только для чётных вариантов).

Лабораторная работа № 2. Программирование алгоритмов разветвляющейся и циклической структуры.

Цель работы. Овладение практическими навыками разработки и программирования вычислительного процесса разветвляющейся и циклических структур, развитие дальнейших навыков по отладке и тестированию программ.

Задания для самостоятельной подготовки.

1. Изучить возможности языка C/C++ для реализации:

- условной передачи управления;
- вычислительного процесса разветвляющейся структуры;
- вычислительных процессов циклической структуры с известным числом повторений в цикле;
- приёма программирования – табулирования функции от одного аргумента (вычисление значений функции при изменении значения аргумента в заданном диапазоне с шагом Δx);
- поразрядных операций.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи.

4. Подготовить тесты (число тестов равно числу ветвей вычислительного процесса) для проверки правильности выполнения программы.

Лабораторная работа № 3. Программирование алгоритмов итерационной циклической структуры.

Цель работы. Овладение практическими навыками разработки и программирования алгоритмов итерационной циклической структуры; приобретение дальнейших навыков по отладке и тестированию программ.

Задания для самостоятельной подготовки.

1. Изучить:
 - организацию итерационных циклов;
 - возможности языка C/C++ для организации таких циклов;
 - приёма программирования – уточнение корня уравнения методом итерации, вычисление суммы членов бесконечного ряда, накопление суммы.
2. Разработать алгоритмы решения задач для двух заданий 1) и 2).
3. Составить программы решения задач для заданий 1) и 2).

Лабораторная работа № 4. Обработка одномерных массивов.

Цель работы. Овладение практическими навыками работы с массивами, особенностями их ввода и вывода; приобретение дальнейших навыков по организации программ циклической структуры с использованием приёмов программирования.

Задания для самостоятельной подготовки.

1. Изучить:
 - способы описания размеров и инициализации массивов, задание динамических массивов;
 - способы ввода и вывода массивом;
 - реализацию приёмов накопления суммы или произведения, запоминание результатов, нахождения наибольшего и наименьшего, пересылка и сдвиг, сортировка;
 - организация вычислительных структур с вложенными циклами.
2. Разработать алгоритм решения в соответствии с заданием.
3. Составить программу решения задачи.
4. Подготовить тексты для проверки программы.

Лабораторная работа № 5. Обработка матриц.

Цель работы. Овладение навыками алгоритмизации и программирования структур с вложенными циклами, навыками использования приёмов программирования во вложенных циклах, способами ввода и вывода матриц, использование динамических массивов.

Задания для самостоятельной подготовки.

1. Изучить:
 - правила организации вложенного цикла с учётом порядка перебора элементов матрицы;
 - правила использования приёмов программирования в структурах с вложенными циклами;
 - способы ввода и вывода матриц;
 - правила организации двумерного динамического массива с помощью операции new.
2. Разработать алгоритм решения в соответствии с заданием.
3. Составить программу решения задачи.
4. Подготовить тестовый вариант программы и исходных данных.

Лабораторная работа № 6. Обработка символьных данных и текстовых файлов.

Цель работы. Овладение навыками алгоритмизации и программирования задач, обрабатывающих символьные данные; ввод данных в файл; чтение данных из файла; использование функций обработки символьных данных.

Задания для самостоятельной подготовки.

1. Изучить:
 - правила записи символьных данных (констант, переменных, массивов и их описание, инициализация);
 - способы ввода и вывода символьных данных;
 - способы обработки символьных данных;

- использование функций обработки символьных данных;
 - стандартные программные решения обработки символов;
 - возможности языка C++ по обработке файла с последовательной организацией: запись данных в файл, чтение из файла, добавление записей в файл, корректировка записей и т.п.
2. Разработать алгоритм решения в соответствии с заданием.
 3. Составить программу решения задачи.
 4. Подготовить тестовый вариант программы и исходных данных.

Лабораторная работа № 7. Программирование с использованием функций пользователя.

Цель работы. Овладение навыками алгоритмизации и программирования задач с использованием функций пользователя различных видов, овладения написанием функций и обращения к ним, выбора параметров функций.

Задания для самостоятельной подготовки.

1. Изучить:

- правила записи (объявления) функций различных видов и способов обращений к ней;
- способы передачи параметров в функцию;
- объявление прототипов функций;
- локальные и статические переменные в функциях;
- выход из функций;
- перегрузка функций;
- функции типа inline.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи.

4. Подготовить тестовый вариант программы и исходных данных.

Лабораторная работа № 8. Программирование с использованием типа данных структурированной.

Цель работы – овладение навыками алгоритмизации и программирования задач с использованием данных типа структур.

Задания для самостоятельной подготовки.

1. Изучить:

- основную терминологию, связанную со структурами, методы доступа к полям структур.
- определения массива структур и указателя на структуру;
- инициализация структур;
- объединение (union);
- использование структур в качестве аргументов функции, передача структур по ссылке, по указателям.

2. Разработать алгоритм решения в соответствии с заданием.

3. Составить программу решения задачи в виде набора функций.

4. Подготовить тестовый вариант программы и исходных данных.

2.2 Среда Delphi и Object Pascal

3 семестр

Лабораторная работа 1. Вычисление функции с помощью разложения в ряд

Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от $x_{нач}$ до $x_{кон}$ с шагом dx с точностью ε . Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента,

значение функции с помощью ряда, с помощью функции библиотеки и количество просуммированных членов ряда.

Лабораторная работа 2. Алгоритмы

Выполнить задания с использованием везде, где это возможно, контейнерных классов и алгоритмов библиотеки.

Лабораторная работа № 3. Работа с реестром.

Прежде чем приступать к программированию необходимо в разделе uses добавить модуль registry. Задание:

Используя пример, создайте программу считывающую и записывающую данные в реестр.

Используя пример, создайте программу, выводящую список программ установленных на компьютере.

Лабораторная работа 4. Вызов API Windows в Delphi

Разработчики библиотеки визуальных компонент (VCL) Delphi очень серьезно поработали над ее проектированием и воплощением в реальность. Как показывает практика, стандартного набора объектов обычно достаточно для создания реальных приложений. И, что более существенно, им (разработчикам) удалось решить очень сложную задачу, стоящую перед создателями любой среды визуального программирования - скрыть от программиста сложность и трудоемкость программирования в Windows и, в то же время, не лишать его возможности доступа к тем богатым возможностям системы, которые предоставляет Windows API.

На примерах показано, как с помощью вызовов Windows API можно управлять объектами из VCL. Кроме того, здесь же можно найти описание некоторых программных трюков, которые помогут придать вашей программе оригинальный вид.

Лабораторная работа № 5. Использование DLL в Delphi

Понятие DLL

Лабораторная работа № 6. Потоки или нити (Thread)

Представьте, что вы хотите вставить в свою программу проверку орфографии. Если вы после каждой нажатой клавиши будете проверять правильность слов, то ваша программа будет очень тормозить. А как же это делает MS Word? Очень просто, после запуска программа запускается поток, который в фоновом режиме производит проверку орфографии. Вы спокойно набираете текст и даже не ощущаете (почти) как параллельный процесс в свободное время производит сложнейшие проверки ваших ошибок.

Лабораторная работа № 7. Хранитель экрана.

Для того, чтобы Delphi мог создать хранитель экрана, Вы должны сделать несколько установок для своей формы:

- Перед объявлением типов вставить вот такую строку: {\$D SCRNSAVE Raver}.
- Свойство формы BorderStyle поменять на bsNone . Это означает, что у формы не должно быть никаких заголовков и обрамлений.
- Все параметры в Border Icons установить в False.
- Свойство формы WindowState поменять на wsMaximized .

- Создать событие OnKeyPress, onMouseDown и onMouseMove и вставить туда всего лишь одну процедуру - Close() .
- На событие FormActivate значения Left и Top нужно установить в ноль.
- Также неплохо было бы на время работы хранителя скрыть курсор мыши, а при выходе снова показать ShowCursor(false), ShowCursor(false) соответственно.

Лабораторная работа № 8. Создание консольных приложений.
Выберите в меню File пункт New[->Other], затем Console Application.

Лабораторная работа № 9. Технология Dock

Для большей ясности – технология Dock - это когда палитру с кнопками можно оторвать от окна и прилепить в другое место или вообще превратить в отдельное окно.

Лабораторная работа № 10. Создание приложений с помощью WinAPI.
Изучение базового шаблона программы для Windows. Принцип отображения информации в окнах.

Лабораторная работа 11. Создание MDI приложений.

Термин MDI (Multiple Document Interface) дословно означает многодокументный интерфейс и описывает приложения, способные загрузить и использовать одновременно несколько документов или объектов. Примером такого приложения может служить диспетчер файлов (File Manager).

Лабораторная работа 12. Использование графики в Delphi

Компонент TForm имеет свойство Canvas, представляющее собой область окна, в которую можно выводить различные графические изображения.

2.3 Применение математических пакетов Mathcad и Matlab для решения задач линейной алгебры, математического анализа и численных методов.

4 семестр.

Лабораторная работа № 1. Использование Mathcad в качестве суперкалькулятора.

Лабораторная работа № 2. Функции. Интервальная переменная.

Определить функцию в Mathcad достаточно просто, для этого необходимо ввести имя функции, в скобках её параметры и оператор присваивания. После чего вводится арифметическое выражение. Затем функция может использоваться наравне с встроенными функциями.

Лабораторная работа № 3. Графики. Вычисление сумм и произведений.

В Mathcad, кроме x-y – графика можно построить и график в полярных координатах, что иногда оказывается очень удобным.

Лабораторная работа № 4. Операции с матрицами.

При решении задач линейной алгебры часто приходится проводить операции с матрицами. При работе с матрицами необходимо использовать панель "Векторы и матрицы" (Vector and Matrix Toolbar).

Лабораторная работа № 5. Дополнительные функции MathCad.

Числовые функции с условиями сравнения:

ceil(x) – наименьшее целое, большее или равное **x**;

floor(x) – наибольшее целое, меньшее или равное **x**;

mod(x,y) – остаток от деления **x/y** со знаком **x**;

функция условных выражений **if**.

Лабораторная работа № 6. Решение переопределенных систем.

Очень часто при решении практических задач приходится решать систему алгебраических линейных уравнений вида $Ax=b$, когда число уравнений (т.е. строчек в матрице **A**) превышает число неизвестных в векторе - столбце **x** (или, что то же самое, число столбцов в матрице **A**). Это получается при достаточно большом наборе наблюдений с переменными параметрами. Когда уравнений и неизвестных немного, обычно убирают "лишние" уравнения - те, которые наиболее сильно отличаются от остальных. При решении больших систем этот метод не подходит - можно выбросить "не то" уравнение. При решения подобных систем используют метод наименьших квадратов для минимизации вектора невязки.

Лабораторная работа № 7. Решение нелинейных уравнений и систем.

Многие уравнения (например, трансцендентные) и системы из них не имеют аналитических решений. Однако они могут решаться численными методами с заданной погрешностью (но не более значения, заданного системной переменной **TOL**).

Лабораторная работа № 8. Программирование.

Хотя **Mathcad** является достаточно мощным средством, с помощью которого можно решить множество задач, существуют нестандартные задачи, не решаемые, не решаемые или решаемые малоэффективно стандартными средствами **Маткада**.

Лабораторная работа № 9. Обыкновенные дифференциальные уравнения.

Почти все функции **Mathcad** предназначены для решения задачи Коши нормальных систем обыкновенных дифференциальных уравнений.

Лабораторная работа № 10. Разложение функций в ряды.

В практике инженерных расчетов часто приходится прибегать к разложению функций в функциональные ряды, т.е. в ряды, составленные из функций определенного вида. Простейшим функциональным рядом является степенной ряд Тейлора.

Лабораторная работа № 11. Решение дифференциальных уравнений при помощи преобразования Лапласа.

Одним из важнейших применений операционного исчисления является решение линейных дифференциальных уравнений с постоянными коэффициентами:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + \dots + a_n y = f(t) \quad (1)$$

где a_0, \dots, a_n – заданные числа (коэффициенты уравнения),

$f(t)$ – заданная функция,

$y(t)$ – искомая (неизвестная) функция.

Требуется решить задачу Коши, т.е. найти решение этого уравнения, удовлетворяющее начальным условиям $y(0) = y_0, y'(0) = y'_0, \dots, y^{(n-1)}(0) = y_0^{(n-1)}$, где $y_0, y'_0, \dots, y_0^{(n-1)}$ – заданные числа.

2.4 Основы программирования трехмерной графики с использованием библиотеки OpenGL

5 семестр

Лабораторная работа 1. Библиотека OpenGL

1. Создайте проект, состоящий из файла с исходным текстом программы 1 и библиотечные файлы OpenGL: `opengl32.lib`, `glu32.lib`, `glaux.lib`. Скомпилируйте и запустите программу. Попробуйте изменить цвет сферы, пользуясь функцией `glColor3f()`
2. "Список трехмерных фигур". С помощью функций, перечисленных в таблице, нарисуйте стандартные фигуры библиотеки GLAUX в два столбца (по пять фигур в каждом столбце): в 1-м – сплошные фигуры, во 2-м – каркасные.
3. "Снеговик". Используя функцию `glRotate`, нарисуйте снеговика. Три сферы, шапка - конус, нос - тоже конус, глаза - сфера, рот можно квадратным сделать - `glBox`.

Лабораторная работа 2. Геометрические примитивы. Общие положения.

Точки, линии, треугольники, четырехугольники, многоугольники - простые объекты, из которых состоят любые сложные фигуры. В предыдущей лабораторной работе мы рисовали сферу, конус и тор и т.д.. OpenGL непосредственно не поддерживает функций для создания таких сложных объектов, т.е. таких функций нет в `opengl32.dll`. Эти функции есть в библиотеки утилит `glu32.dll`, и устроены они следующим образом. Для того чтобы нарисовать сферу функция `auxSolidSphere` использует функции из библиотеки `glu32.dll`, а те в свою очередь, используют базовую библиотеку `opengl32.dll` и из линий или многоугольников строят сферу.

Лабораторная работа 3. Полигональная аппроксимация поверхностей. Векторы нормали.

Вектором нормали к поверхности в данной точке называется вектор перпендикулярный этой поверхности. Векторы нормали определяют ориентацию поверхности в пространстве, в частности её расположение относительно источников света. Эти векторы OpenGL использует для расчётов количества света, получаемого объектом в его вершинах.

Нормаль может быть одинаковой в нескольких вершинах. В каких-либо иных точках, кроме вершин, OpenGL задавать вектор нормали не позволяет. Для указания вектора нормали используется одна из функций `glNormal*()`.

Лабораторная работа 4. Цвет и освещение.

Цветовая модель RGB.

На мониторе компьютера видимые цвета формируются путём смешивания трёх основных цветов: красного (Red), зелёного (Green) и синего (Blue). В памяти компьютера значение цвета хранится в виде трёх чисел - компонентов R,G и B. К ним иногда добавляется четвёртый компонент A (прозрачность).

В OpenGL текущий цвет задается функциями `glColor*()`.

В OpenGL двумерное изображение для вывода на экран хранится в виде массива значений цветов, соответствующих каждому пикселю изображения. Этот массив называется цветовым буфером. Значения компонентов R,G и B лежат в диапазоне от 0.0 (минимальная интенсивность) до 1.0 (максимальная интенсивность). Если в OpenGL включено освещение, то для вычисления цвета пикселя в цветовом буфере производится ряд вычислений, в которых участвует цвет примитива, например, красный мяч в ярком синем цвете будет выглядеть иначе, чем при белом цвете.

Лабораторная работа 5. Задание полигональных моделей объектов. Трёхмерная визуализация с использованием OpenGL.

Цель работы: ознакомиться с основным способом задания полигональных моделей – методом тиражирования сечений; ознакомиться со средствами трёхмерной визуализации в OpenGL (источники света, свойства материалов, текстуры).

2.5 Динамические типы данных и объектно-ориентированное программирование.

6 семестр

Цели лабораторных работ:

- изучение наиболее распространенных алгоритмов решения задач с использованием сложных структур данных.

Содержание работ одинаково для всех лабораторных работ.

Язык программирования – язык Си/Си++.

Порядок выполнения работы

1. Изучить необходимый материал по теме лабораторной работы, руководствуясь методическими указаниями к теме.
2. Получить допуск к работе, ответив на контрольные вопросы.
3. Разработать структуру данных для представления основных исходных данных и/или результатов для выданного преподавателем варианта задания.
4. Разработать алгоритм и написать программу решения задачи.
5. Подготовить набор тестов и отладить программу.
6. Оформить отчет по лабораторной работе. Отчет по работе включает следующие пункты:
 1. Условие задачи
 2. Анализ задачи
 3. Структура основных входных и выходных данных
 4. Алгоритм решения задачи
 5. Текст программы
 6. Набор тестов
 7. Результаты отладки и их анализ
 8. Защитить лабораторную работу. Защита состоит в обсуждении выбранной структуры данных и алгоритма решения задачи, ответе на контрольные вопросы, решении контрольных задач.

Лабораторная работа №1. Линейные однонаправленные списки

Цель работы: изучить тип указатель; получить навыки в организации и обработке однонаправленных списков.

Лабораторная работа № 2. Линейные двунаправленные списки

Цель работы: получить практические навыки организации двунаправленных (двусвязных) списков и их использования при решении задач.

Методические указания

Динамический список, в котором каждый элемент (кроме возможно первого и последнего) связан с предыдущим и следующим за ним элементами называется двусвязным. Каждый элемент такого списка имеет два поля с указателями: одно поле содержит ссылку на следующий элемент, другое поле – ссылку на предыдущий элемент и третье поле - информационное. Наличие ссылок на следующее звено и на предыдущее позволяет двигаться по списку от каждого звена в любом направлении: от звена к концу списка или от звена к началу списка, поэтому такой список называют еще и двунаправленным.

Лабораторная работа № 3. Структуры данных стеки и очереди.

Цель работы: сформировать практические навыки организации таких распространенных структур как стеки и очереди и их использования при решении задач.

Методические указания

Стек – это список, у которого доступен один элемент (одна позиция). Этот элемент называется вершиной стека. Взять элемент можно только из вершины стека, добавить элемент можно только в вершину стека.

Очередь – это список, у которого доступны два элемента (две позиции) начало и конец очереди. Поместить элемент можно только в конец очереди, а взять элемент только из ее начала.

Стеки и очереди могут быть организованы различными способами. При этом они могут быть размещены как в статической памяти, так и в динамической.

Статический стек можно задать следующим образом:

```
struct stack
{ int top;
  type data [ n ] ;}
```

Здесь type – тип элементов, хранимых в стеке. Поле top определяет вершину стека. Ограничение на количество элементов в массиве (константа n) задает размер стека. Статический стек может быть полон.

Лабораторная работа № 4. Структуры данных: деревья.

Цель работы: изучить основные алгоритмы работы с деревьями; получить практические навыки разработки и использования этих структур и алгоритмов для решения задач.

Лабораторная работа № 5. Управление таблицами.

Цель работы: получить практические навыки организации таблиц, обработки таблиц и их использования при решении задач.

Методические указания

Таблица – это список состоящий из конечного множества элементов, при чем каждый элемент характеризуется рядом признаков (свойств). Один из признаков, называемый ключом, позволяет отличить один элемент от другого (идентифицировать элемент). Ключ может однозначно определять элемент таблицы (ключи всех элементов различны) или неоднозначно (в таблице есть элементы с равными ключами).

Все действия над элементами выполняются в соответствии с их ключами: по ключу элементы выбираются из таблицы и добавляются в нее.

Таблицы можно хранить как в статической, так и в динамической памяти. В данной работе рассматриваются статические таблицы. Статическая таблица отображается массивом.

Лабораторная работа № 6. Упорядочение данных

Цель работы: изучить основные алгоритмы упорядочения (сортировки) данных.

Методические указания

Сортировкой называется процесс (операция) перегруппировки объектов в некотором определенном порядке. Различают внутреннюю сортировку и внешнюю.

Алгоритмы внутренней сортировки упорядочивают данные, хранимые в оперативной памяти, алгоритмы внешней сортировки обрабатывают данные, хранимые во внешней

памяти. В лабораторной работе № 6 рассматриваются алгоритмы внутренней сортировки данных, хранимых в статических таблицах.

Лабораторная работа 7. Наследование

Примечание: во всех классах должно быть хотя бы одно поле, память под которое выделяется динамически.

Лабораторная работа 10. Контейнерные классы.

Выполнить задания с использованием соответствующих контейнерных классов библиотеки.

IV. КОНТРОЛЬ ЗНАНИЙ

3.1 Фонд тестовых и контрольных заданий для оценки качества знаний по дисциплине

Инструкция: все задания имеют одну и ту же форму – с выбором одного ответа из четырех предложенных. Заданиями с 4-го по 26-е проверяется знание языка программирования Object Pascal среды программирования Borland Delphi. Текст на русском языке, заключенный в символы < и >, является смысловым описанием некоторой конструкции языка программирования. Время тестирования – 60 минут.

1. В конфигурацию компьютерной системы не входит...

- 1) кошка 2) системный блок 3) монитор 4) мышь

2. Устройством управления компьютером манипуляторного типа является...

- 1) монитор 2) сканер 3) мышь 4) системный блок

3. Внутри системного блока не находится...

- 1) принтер 2) материнская плата 3) видеокарта 4) процессор

4. Комментарием является последовательность символов, стоящих правее пары символов...

- 1) => 2) >> 3) // 4) ||

5. Буквы русского алфавита могут использоваться в...

- 1) идентификаторах 2) символьных выражениях
3) зарезервированных словах 4) директивах компилятора

6. Идентификатором может быть последовательность символов...

- 1) чай 2) -d- 3) сAT 4) 7id

7. Раздел описания (объявления, декларации) чего-либо оканчивается символом...

- 1) ; 2) ,
3) : 4) _

8. Функция pred возвращает...

- 1) символ, код которого равен значению аргумента функции
- 2) код символьного значения аргумента функции
- 3) значение порядкового типа, имеющее порядковый номер на единицу больший, чем порядковый номер значения аргумента функции
- 4) значение порядкового типа, имеющее порядковый номер на единицу меньший, чем порядковый номер значения аргумента функции

9. Раздел описания (объявления, декларации) типов начинается словом...

- | | |
|----------|----------|
| 1) var | 2) type |
| 3) const | 4) label |

10. Ввести данные с клавиатуры позволяет процедура...

- | | |
|---------|----------|
| 1) get | 2) write |
| 3) read | 4) put |

11. Символьным является выражение...

- | | |
|------------------|---------------|
| 1) ord(false) | 2) sin(5.6)<7 |
| 3) 'dog'+ ' cat' | 4) chr(33) |

12. Логическим является тип...

- | | |
|-----------|-------------|
| 1) real | 2) boolean |
| 3) string | 4) ansichar |

13. Результатом операции not not not (3>7) является значение...

- | | |
|------------|---------|
| 1) false | 2) 7 |
| 3) 'false' | 4) true |

14. К порядковому типу не относится тип...

- | | |
|------------|------------|
| 1) integer | 2) char |
| 3) real | 4) boolean |

15. Составной оператор ограничен...

- | | |
|--------------------|------------------------|
| 1) символами { и } | 2) символами : и . |
| 3) символами < и > | 4) словами begin и end |

16. Пустой оператор содержится во фрагменте...

- | | |
|---------------------|------------|
| 1) begin y:=5; end. | 2) y:=0; |
| 3) writeln(' '); | 4) y:=5<2; |

17. В операторе выбора значение выражения, стоящего после слова case, не может относиться к типу...

- | | |
|------------|------------|
| 1) real | 2) boolean |
| 3) integer | 4) char |

18. Если в операторе цикла с параметром (оператор for) используется слово to, то при выполнении очередной итерации порядковый номер значения счетчика (параметра цикла)...

- 1) уменьшается на единицу
- 2) увеличивается на единицу
- 3) уменьшается на заданное значение
- 4) увеличивается на заданное значение

19. Дан оператор цикла с постусловием:

repeat <операторы> until <логическое выражение>;

Начиная со второй итерации, тело цикла выполняется оператором цикла, если значение логического выражения...

- 1) true
- 2) false
- 3) 'false'
- 4) 'true'

20. Нет категории файла...

- 1) графический файл
- 2) текстовый файл
- 3) типизированный файл
- 4) нетипизированный файл

21. Группу формальных параметров – результатов предваряют словом...

- 1) const
- 2) var
- 3) out
- 4) in

22. В описании метода указать на объект, вызвавший этот метод можно словом...

- 1) inherited
- 2) self
- 3) me
- 4) object

23. Какие методы не могут быть перекрыты методами классов-наследников...

- 1) виртуальные
- 2) абстрактные
- 3) статические
- 4) динамические

24. При описании типа-записи используют слово...

- 1) array
- 2) set
- 3) record
- 4) class

25. Тип индекса у массива не может относиться к...

- 1) типу вещественных чисел
- 2) перечислимому типу
- 3) интервальному типу
- 4) логическому типу

26. Операцию пересечения множеств обозначают символом...

- 1) –
- 2) +
- 3) *
- 4) ~

27. В программе Mathcad перед набором индексов элемента массива с клавиатуры вводится символ...

- 1) (
- 2) {
- 3) <
- 4) [

28. В программе Mathcad значение нижней границы индексации элементов массива определяется переменной...

- | | |
|----------------|----------|
| 1) origin | 2) index |
| 3) matrixindex | 4) lim |

29. В программе Mathcad для реализации нелинейной регрессии общего вида используют функцию(ии)...

- | | |
|----------------------|-----------|
| 1) intercept и slope | 2) linfit |
| 3) regress и interp | 4) genfit |

3.2 Вопросы к зачету.

2 семестр

1. Понятие о файлах. Процедуры открытия и закрытия файлов.
2. Процедуры и функции работы с файлами.
3. Текстовые файлы.
4. Записи. Записи с вариантами.
5. Процедурные типы.
6. Не типизированные параметры процедур и функций.
7. Какие типы величин используются в языке C++?
8. Указать диапазон значений величин типов int и float.
9. Какие имена переменных допустимы в программе?
10. Можно ли в качестве операнда в арифметическом выражении использовать: а) имя стандартной функции; б) имя массива; в) имя символьной переменной или переменной логического типа?
11. Что напечатает следующий оператор cout << “*\n**\n***\n****\n*****\n”?
12. Перечислить действия, реализуемые при выполнении условного оператора.
13. Что такое вычислительный процесс разветвляющейся структуры?
14. Как организовать разветвление вычислений а) на две ветви; б) на три ветви?
15. Зачем необходимо при отладке программы тестировать все ветви алгоритма?
16. Указать последовательность действий, выполняемых при организации циклических участков программы с заданным числом повторений
17. Что такое итерационный циклический процесс? Его отличия от цикла с заданным числом повторений.
18. Какие условия сходимости метода итераций?
19. Каково условие выхода из цикла при вычислении значения суммы бесконечного ряда?
20. Зачем используются рекуррентные соотношения для вычисления значений ряда?
21. Какие операторы можно использовать для описания массивов?
22. Возможен ли выход из внутреннего цикла до его полного завершения, досрочное завершение работы программы?
23. В каких случаях можно не освобождать память, выделенную посредством new []?
24. Указать основные правила организации вложенных циклов?
25. Как добавить элемент в односвязный список?
26. Основные отличия динамического выделения памяти, от статического?

3 семестр

1. Структурные типы данных в Delphi.
2. Описание нового типа.
3. Сложные структуры данных.
4. Основные стандартные функции для работы с типами.
5. Преобразование типов.
6. Инициализация констант сложных типов.
7. Структура подпрограмм.

8. Понятие объекта.
9. Понятие класса.
10. Типы методов.
11. Динамическое конструирование объектов.
12. Использование визуальных компонентов.
13. Компонент Меню.
14. Компонент Контекстное меню.
15. Компонент Текстовая область.
16. Компонент Переключатель.
17. Компонент Группа переключателей.
18. Компонент Поле со списком.
19. Компонент Флажок.
20. Компонент Полоса прокрутки.
21. Иерархия компонентов.
22. Класс Tobject.
23. Класс Tpersistent (наследник Tobject).
24. Класс Tcomponent (наследник Tpersistent).
25. Форма.
26. Управление проектом. Добавление новой формы.
27. Что такое отладка программы?
28. Причины ошибок.
29. Синтаксические ошибки.
30. Логические ошибки.
31. Выполнение программы по шагам.
32. Просмотр и изменение значений.
33. Просмотр и анализ кода.
34. Прерывания по условию.
35. Организация точек прерывания в группы.
36. Ведение протокола работы.
37. Отладка внешних процессов.
38. Инспектор объектов.
39. Генерация исключительных ситуаций.
40. Стандартные классы исключительных ситуаций.
41. Контроль над исключительными ситуациями.
42. Программный обработчик ошибок.

4 семестр

1. МАТНСАД. Основные возможности пакета. Создание и редактирование формул. Создание математических и текстовых областей, параграфы.
2. Символьные вычисления. Графические возможности. Создание различного рода графиков, их форматирование и использование.
3. Форматирование выражений и результатов. Локальный и глобальный формат. Дискретный аргумент. Переменные и константы. Способы определения и вычисления переменных и функций. Встроенные переменные.
4. Создание векторов и матриц, способы их отображения. Оператор векторизации. Основные встроенные функции. Решение СЛАУ.
5. Аппроксимация функций.
6. Решение алгебраических уравнений.
7. Программирование.
8. Решение дифференциальных уравнений и систем.
9. Ряды Фурье.
10. Задачи из теории вероятностей.

11. Решение задач численными методами.
12. Программирование: составные операторы, условные операторы, условные циклы.
13. Ввод и вывод данных в файлы и в другие программы.
14. Matlab. Переменные и их типы. Команды для задания различного вида матриц.
15. Matlab. Построение графиков функций, виды графиков.
16. Matlab. Нахождение корней уравнения, полинома.
17. Matlab. Многомерные массивы: определение, команды обработки.
18. Matlab. Программирование: m-файлы и m-функции.
19. Matlab. Решение уравнений, задач мат. анализа и линейной алгебры с помощью тулбоксов.

5 семестр

1. Назначение библиотеки OpenGL.
2. Основные возможности OpenGL.
3. Имена функций OpenGL.
4. Системы координат.
5. Служебные графические операции.
6. Свойство точек, отрезков, многоугольников.
7. Полигональная аппроксимация поверхностей.
8. Цветовая модель RGB.
9. Задание способа закраски.
10. Освещение.
11. Задание свойства материала.
12. Смещение цветов и прозрачность.
13. Растровые объекты: изображения и текстуры.
14. Полигональная аппроксимация поверхностей.
15. Геометрические примитивы.
16. Генерация движущихся изображений.

6 семестр

1. Создание пользовательского модуля и использование его в программе.
2. Структурированные типы данных. Множества.
3. Динамические данные. Указатели.
4. Ссылочные типы.
5. Динамические массивы.
6. Списки.
7. Стек. Организация стека.
8. Операции со стеками. Удаление и добавление элемента в стек.
9. Выделение и освобождение динамической памяти. Процедуры и функции, используемые при работе с указателями и динамической памятью.
10. Очередь. Организация очереди.
11. Удаление первого и добавление последнего элемента в очередь.
12. Принципы объектно-ориентированного программирования.
13. Описание объекта.
14. Инкапсуляция. Методы. Классы и объекты.
15. Сообщения и события, реагирование объекта на сообщения и события.
16. Наследование.
17. Полиморфизм.
18. Иерархия объектов.
19. Виртуальные методы. Объявление виртуальных методов.
20. Конструкторы и деструкторы.
21. Понятие о функциях Windows API.